On the Implications of Delay Adaptability for Learning in Pulsed Neural Networks*

Michael Schmitt†

Lehrstuhl Mathematik und Informatik, Fakultät für Mathematik
Ruhr-Universität Bochum, D–44780 Bochum, Germany
mschmitt@lmi.ruhr-uni-bochum.de

ABSTRACT

We consider a model for networks of neurons that compute and communicate in terms of pulses. In addition to weights and thresholds, which are commonly the parameters of artificial neural networks, these pulsed neural networks have adaptable delays. We present and discuss the most recent and prominent results on the complexity of computing and learning using this new type of parameter.

INTRODUCTION

Any kind of learning is undoubtedly based on some plasticity in which observations and experiences leave their traces. In artificial neural networks such plasticity is commonly represented by adjustable weights characterizing the strengths or efficacies of the connections between the neurons. During past years the development of learning methods for artificial neural networks has lead to a large body of algorithms for tuning the weights of these networks (see, e.g., (Bishop, 1995)). As far as the biological relevance of these artificial networks is concerned, however, there is wide agreement among neural network researchers that these models are oversimplistic representations of their natural counterparts. Among the many issues that are not captured by these artificial models there is at least one that is now believed to play a major role for computing and learning in biological neural systems: the timing of neural activity.

Neural activity takes place in terms of action potentials, or spikes, which are short pulses that neurons send to other neurons via synaptic connections. From numerous neurobiological experiments it has become increasingly clear that information cannot only be encoded in the firing rate of neurons but must also be contained in the temporal pattern of their firing. Some of the most striking results showing that rate coding alone is not sufficient come from the work of (Rieke et al., 1997). By studying a motion-sensitive neuron in the blowfly they found that the input stimulus could be reconstructed from the activity pattern of this neuron which was then shown to require a high timing precision on the order of one millisecond. Moreover, these researchers concluded that the frequency of firing of this neuron is too low in order for the subsequent neurons to make use of firing rate information. Similar inferences have been drawn by (Thorpe et al., 1996) who made experiments with human subjects performing a visual pattern recognition task. Measuring the performance time and estimating the number of synaptic stages involved they argued that the sampling interval for each neuron must be rather short and therefore the fast computation that is required for this task is unlikely to be based on the firing rate. (Recce, 1999) describes and discusses further experimental results in this direction.

*Work supported in part by the ESPRIT Working Group in Neural and Computational Learning II, NeuroCOLT2, No. 27150.

†http://www.ruhr-uni-bochum.de/lmi/mschmitt/

Conventional neuron models which are widely used in machine learning tasks, such as the McCulloch-Pitts neuron or threshold gate (McCulloch and Pitts, 1943) with the Perceptron learning rule (Rosenblatt, 1958) or sigmoidal neural networks with the backpropagation algorithm (Bishop, 1995), cannot describe neural computation in terms of single spikes since they have been designed for other purposes. Therefore, in order to be able to understand biological information processing capabilities, to reproduce them and reach their performance by artificial neural networks one has to build networks consisting of biologically more realistic neuron models. Among these is a class of neuron models called spiking neurons, integrate-and-fire neurons, or pulsed neurons (Gerstner, 1999; Koch, 1999; Maass, 1997).

Pulsed neural networks prove to be adequate when computation in neural systems is interpreted as being based not only on average firing activity but also on the timing of single action potentials. These network models have now gained notable interest since on the one hand they are supported by results from neurobiological experiments such as described above that cannot be explained in terms of rate coding, on the other hand they have turned out to be suitable for hardware implementations of neural networks in analog VLSI. Attention from researchers in both of these areas has manifested itself, for instance, in a recent book (Maass and Bishop, 1999).

Obviously, when computation is based on the timing of pulses the transmission delay, that is, the time that a pulse needs to travel along a connection from one neuron to the other, becomes critical. Therefore, in pulsed neural networks these delays are essential parameters that must be taken into account and included as adjustable parameters. There are many mechanisms known in neurobiology for the modification of delays between neurons. One such mechanism, for instance, is the selection of a few synapses from an initially large set of multiple synapses with different delays. Some other mechanisms are discussed in (Agmon-Snir and Segev, 1993; Gerstner et al., 1996; Gerstner et al., 1999). Furthermore, learning rules for delay adaptation in artificial neural networks have also been designed and studied using numerical simulation experiments by several researchers, for instance (Bodenhausen and Waibel, 1991; Eurich et al., 1997; Hüning et al., 1998; Natschläger and Ruf, 1998; Steuber and Willshaw, 1997).

Naturally, the question arises whether the computational task of delay adaptation requires a similar amount of resources, e.g. in terms of the number of training examples or computing time, as weight adaptation, or whether it is a substantially different problem. In this paper we approach this question from a theoretical point of view. We consider a simple model for a pulsed neuron that has connections parameterized by weights and transmission delays. We present and discuss recent results on the complexity of computing and learning with pulsed neurons and networks using adaptable delays. We do not give any proofs here but refer the interested reader to the original papers where the results appeared first.

The paper is organized as follows: In the following section we introduce the definition for the model of a pulsed neuron. Then we explore its computational capabilities by considering what types of Boolean functions it can realize. The results show that a much richer class of functions can be computed by a pulsed neuron than by a McCulloch-Pitts neuron. The next two sections then present bounds for the sample complexity of learning, that is, for the number of examples required for good generalization when using pulsed neurons and networks, respectively. These bounds are given in terms of a combinatorial parameter called Vapnik-Chervonenkis (VC) dimension which is known to yield tight bounds for the sample complexity for so-called probably approximately correct learning (Blumer et al., 1989; Valiant, 1984; Vidyasagar, 1997). These sample complexity bounds show that tuning delays turns out to be significantly more complex than tuning weights. Finally we consider the computational complexity of learning with adaptable delays presenting an intractability result for pulsed neurons showing that efficient algorithms for tuning delays are unlikely to exist. We conclude with some final remarks.

Notation. Several results presented in this paper are formulated in terms of upper or lower bounds. In order to abstract from constant factors we use O-notation for describing the asymptotic behaviour of these bounds. Given functions f and g, we say f is $O(g)$ if $f(n) \leq c \cdot g(n)$ for some

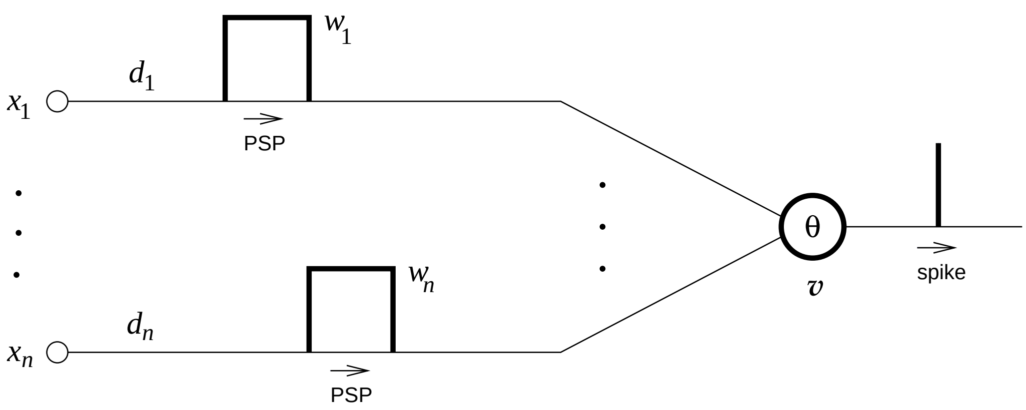

Figure 1: The model for a pulsed neuron v. A spike (not shown) arriving from a presynaptic (or input) neuron x_i generates a postsynaptic potential (PSP) in form of a pulse of height w_i and unit length after a delay of d_i (see Equ. (1)). As soon as the membrane potential (defined in Equ. (2)) reaches the threshold θ (see Equ. (3)), v emits a spike itself.

constant c and all sufficiently large n. Further, we say f is $\Omega(g)$ if g is $O(f)$, and we say f is $\Theta(g)$ if f is both $O(g)$ and $\Omega(g)$.

PULSED NEURONS AND NETWORKS

In a *pulsed neural network* each node v receives inputs in the form of short pulses, also known as spikes, from its presynaptic neurons $x_1, \ldots, x_n$ (see Figure 1). Each connection between two neurons is parameterized by a weight $w_i \in \mathbb{R}$ and a delay $d_i \in \mathbb{R}^+$ (where $\mathbb{R}^+ = \{x \in \mathbb{R} : x \geq 0\}$). One usually distinguishes between axonal, synaptic, and dendritic delays. We assume here that d_i incorporates all three of these delays. If v receives a spike that has been emitted by some presynaptic node x_i at some time t_i, say, this will generate a rectangular pulse in v described by the time-dependent function $h_i(t - t_i)$ with

$$h_i(t) = \begin{cases} 0\ , & t < d_i \text{ or } t \geq d_i + 1\ , \\ w_i\ , & d_i \leq t < d_i + 1 \end{cases} \tag{1}$$

where continuous time is denoted by the variable t. This function serves as a simple approximation of the so-called postsynaptic potentials of biological neurons. The *membrane potential* describes the internal continuous-time state of v and is defined by

$$P_v(t) = \sum_{i \in F} h_i(t - t_i) \tag{2}$$

where F is the set of numbers of those presynaptic neurons which have emitted a spike. As soon as P_v reaches a certain threshold θ_v, v emits a spike. More precisely, the *firing time* t_v of v is defined by

$$t_v = \min\{t : P_v(t) \geq \theta_v\} \tag{3}$$

if such a minimum exists.

We consider two types of coding on which computations of pulsed neurons can be based. *Binary coding* represents Boolean values assuming that a neuron fires at some fixed time if this firing encodes a 1, and that the neuron does not fire at all if it encodes a 0. For *analog coding* the firing of a neuron at time t_v represents the real number t_v. A similar way of coding values in terms of action potential timing has been studied by (Hopfield, 1995).

Neuron models of the type considered here are also referred to as spiking neurons, or integrate-and-fire neurons. In particular, the model defined above has first been introduced in (Maass, 1997). For simplicity, we call it a pulsed neuron throughout the paper. Compared to conventional neuron models, pulsed neurons are not only a further step towards a more biologically plausible description of neural systems, they are but also suitable for implementations of pulsed neural networks in analog VLSI. In particular, pulses of the shape considered here can easily be generated on hardware as explained in (Murray, 1999). Further discussions of this and more general types of neuron models can be found in the surveys (Gerstner, 1999; Maass, 1999).

COMPUTATIONAL POWER OF PULSED NEURONS

In this section, we restrict the inputs to binary values and ask how many and what types of Boolean functions can be computed by a pulsed neuron. Computation of Boolean functions by single neurons has also been considered in the seminal work of (McCulloch and Pitts, 1943). Thus we are able to directly compare the computational power of pulsed and McCulloch-Pitts neurons. First we show how many out of the 2^{2^n} Boolean functions on n variables a pulsed neuron can compute.

Theorem 1 (Maass and Schmitt, 1997) *There are $2^{\Theta(n^2)}$ many Boolean functions computable by a pulsed neuron with binary coding of the inputs.*

This quantitative estimate does not reveal what types of functions can be computed. In order to gain more insight we consider two well-established function classes. A Boolean function f is linearly separable if it can be computed by a McCulloch-Pitts neuron, or synonymously a threshold gate, which means that there exist weights $w_1, \ldots, w_n$ and a threshold t such that

$$\begin{aligned} & f(x_1, \ldots, x_n) = 1 \\ & \quad \Longleftrightarrow \quad w_1 x_1 + \cdots + w_n x_n \geq t. \end{aligned}$$

A Boolean formula is in $\mu-$DNF if can be written in disjunctive normal form such that each variable occurs at most once. As an example consider the formula

$$x_1 x_2 \vee x_3 x_4 \vee \cdots \vee x_{n-1} x_n.$$

It turns out that this particular function cannot be computed by a McCulloch-Pitts neuron. However, $\mu-$DNF formulas are appropriate to model a biologically relevant neural behaviour known as coincidence detection. This behaviour refers to the firing of a neuron in response to input spikes that are precisely related in time (Koch, 1999). We show in particular that a pulsed neuron can compute any $\mu-$DNF formula.

Theorem 2 (Maass, 1997; Maass and Schmitt, 1997) *A pulsed neuron with binary coding of the inputs can compute any linearly separable Boolean function. Further, it can compute any Boolean function that can be written as a $\mu-$DNF formula. Both inclusions are strict.*

The complexity of a Boolean function can be characterized in terms of the so-called *threshold number* of the function. This is defined as the smallest number k such that the function can be computed by a disjunction of k threshold gates. The threshold number of a Boolean function can range from 0 to 2^{n-1}, where n is the number of variables, and is 1 for a linearly separable Boolean function. It provides a way to characterize Boolean functions in terms of linear separability. The following result shows that the computational power of a single pulsed neuron comes up to that of a small neural network of threshold gates with one hidden layer.

Theorem 3 (Schmitt, 1998) *The threshold number of a pulsed neuron with n inputs is $\Theta(n)$.*

Another characterization of the complexity of a Boolean function is provided by its *threshold order*. When considering separability of Boolean functions by polynomials instead of linear functions, the question arises how large the degree of a polynomial must be to achieve this separability. The threshold order of a Boolean function is the smallest such degree. This number can range from 0 to n, obviously being 1 for linearly separable Boolean functions. The following result gives a lower bound for a pulsed neuron showing that the order is not far from the maximally possible value n.

Theorem 4 (Schmitt, 1998) *The threshold order of a pulsed neuron with n inputs is $\Omega(n^{1/3})$.*

Neuron models that compute polynomially separable Boolean functions are also known as polynomial threshold gates, sigma-pi units, or higher-order neurons (see, e.g., (Softky and Koch, 1995)). Several biophysical mechanisms are known that mimic higher-order interactions among inputs described by these neuron models (Koch and Poggio, 1992; Koch, 1999). The result stated in Theorem 4 implies that for the simulation of a pulsed neuron by a higher-order neuron the degree cannot be bounded in advance but must grow as the number of inputs increases.

Corollary 5 (Schmitt, 1998) *There is no higher-order neuron with fixed degree that can simulate a pulsed neuron.*

SAMPLE COMPLEXITY FOR PULSED NEURONS

An important characterization of the complexity of learning using a specific hypothesis class to represent the final result is provided by the sample complexity of the class, that is, the number of examples that a learning algorithm needs to select hypotheses from this class that generalize well. In this section we consider the sample complexity for a single pulsed neuron with adapable delays, weights, and threshold. We provide estimates for the Vapnik-Chervonenkis dimension and the pseudo dimension. These dimensions are not only known to yield good bounds on the sample complexity for learning, they also represent the computational variety of the particular function class under consideration, which is here the class of functions that can be computed by a pulsed neuron. The VC dimension for a pulsed neuron has also been considered by (Zador and Pearlmutter, 1996). In their model, however, the only adaptable parameters are the membrane time constant and the threshold whereas the delays are assumed to be fixed.

We now give the definitions of VC and pseudo dimension. We call a partition of a set $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_m\} \subseteq \mathbb{R}^n$ into two disjoint subsets S_0, S_1 a *dichotomy*. If $\mathcal{F}$ is a set of $\{0,1\}$-valued functions we say that $\mathcal{F}$ *shatters* S if for each dichotomy S_0, S_1 of S there is some $f \in \mathcal{F}$ that *induces* this dichotomy, i.e., that satisfies $f(S_0) \subseteq \{0\}$ and $f(S_1) \subseteq \{1\}$. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{F}$ is defined as the cardinality of the largest set that is shattered by $\mathcal{F}$. For a set $\mathcal{F}$ of real-valued functions, S is said to be *P-shattered* by $\mathcal{F}$ if there exist real numbers $y_1, \ldots, y_m$ such that every dichotomy of $\{(\mathbf{s}_i, y_i) : i = 1, \ldots, m\}$ is induced by some function of the form $(\mathbf{s}, y) \mapsto \mathrm{sign}(f(\mathbf{s}) - y)$ for some $f \in \mathcal{F}$. The *pseudo dimension* of $\mathcal{F}$ is the cardinality of the largest set P-shattered by $\mathcal{F}$. Finally, we define the VC dimension (pseudo dimension) of a pulsed neuron with binary (analog) coding of the output as the VC dimension (pseudo dimension) of the set of functions computed by the neuron with all possible assignments of values to its parameters (i.e. delays, weights, and threshold).

It is well known that the VC dimension of a McCulloch-Pitts neuron is equal to the number of adaptable parameters (i.e. weights and threshold) and hence $n+1$ if n is the number of inputs. The following result shows that this dimension grows super-linearly when delays instead of weights become adjustable.

Theorem 6 (Maass and Schmitt, 1997) *The VC dimension of a pulsed neuron with n adaptable delays is $\Omega(n \log n)$. This holds even if the inputs are restricted to binary values and all weights are kept fixed.*

The following corresponding upper bound holds even if we allow the weights to be adjustable and the input values to be from the reals.

Theorem 7 (Maass and Schmitt, 1997) *The VC dimension of a pulsed neuron with n analog coded inputs and binary coded output is $O(n \log n)$.*

Thus both results establish a tight bound for the VC dimension of a pulsed neuron.

Corollary 8 (Maass and Schmitt, 1997) *A pulsed neuron has VC dimension $\Theta(n \log n)$. This holds for binary as well as for analog coding of the inputs.*

So far we have restricted our analysis to binary output values. Clearly, artificial neural networks are also used for the computation and approximation of real-valued functions. A well-known neuron model for the computation of real-valued functions is the sigmoidal gate. It computes its output values by applying the standard sigmoid function $\sigma(y) = 1/(1 + e^{-y})$ to the weighted sum of the input values. We now compare the sigmoidal gate with the pulsed neuron in terms of the pseudo dimension. As in the case of the VC dimension for the McCulloch-Pitts neuron, the pseudo dimension of the sigmoidal gate is equal to the number of parameters. Again, for a pulsed neuron computing real-valued functions we have a super-linear bound.

Theorem 9 (Maass and Schmitt, 1997) *The pseudo dimension of a pulsed neuron with n analog coded inputs and analog coded output is $\Theta(n \log n)$.*

SAMPLE COMPLEXITY FOR PULSED NETWORKS

We consider feedforward neural networks which is the type of neural architecture most widely used in applications. Since the results in this section are easily transferred to networks with multiple output nodes it is convenient to assume that there is just one output node. Binary and analog coding may be independently used to encode values as inputs to or outputs from the network. In correspondence with the previous section we define the VC dimension of a pulsed neural network as the VC dimension of the function class arising from all possible parameter assignments to the network. In the following, we present upper and lower bounds for the VC dimension. (When considering networks, corresponding results for the pseudo dimension can easily be derived in the case of upper bounds. For lower bounds they immediately follow from the definition.)

It is well known that a network of threshold gates with C connections, where each connection has an adjustable weight and each neuron an adjustable threshold, has VC dimension $O(C \log C)$. Further, by the construction of neural networks with superlinear VC dimension this bound has been shown to be asymptotically tight (see, e.g., (Vidyasagar, 1997) for these and other results concerning neural networks with adaptable weights). The following result shows that there is a significantly larger lower bound for the case of adaptable delays in pulsed neural networks.

Theorem 10 (Maass and Schmitt, 1998) *For each n a pulsed neural network with $O(C)$ connections can be constructed that has VC dimension $\Omega(C^2)$. This even holds if only the delays are adaptable and all weights and thresholds remain fixed.*

In (Maass and Schmitt, 1998) this result is shown for inputs from the Boolean domain and for a network using binary coding of the inputs. Interestingly, this quadratic lower bound equals the best known lower bound for sigmoidal neural networks. For sigmoidal networks, however, only an upper bound of $O(C^4)$ is known thus far. The following result, which holds even if we allow delay and weight adaptability and even if the inputs are real numbers, shows that the quadratic lower bound is asymptotically tight in the case of pulsed neural networks.

Theorem 11 (Maass and Schmitt, 1998) *Consider a pulsed neural network where all delays, weights, and thresholds are adaptable, and let C be the number of connections. The VC dimension of this network is $O(C^2)$ even for analog coding of the inputs.*

The construction for the lower bound of Theorem 10 requires the depth of the network (that is, the length of the longest path leading from an input node to the output node) to grow with the number of input nodes. Such networks of unrestricted depth, however, are rarely used in practice. It appears more adequate to consider small-depth networks where the neurons are arranged in layers, with connections only between subsequent layers. The following result shows that the VC dimension of such a multi-layered network of pulsed neurons is almost linear in terms of the number of connections and the number of layers.

Theorem 12 (Schmitt, 1999) *A multi-layer pulsed neural network with C connections and L layers has VC dimension $O(CL\log(CL))$.*

If the number of layers is considered a constant then this upper bound appears to be asymptotically optimal. This follows from Theorem 6 where the lower bound $\Omega(n\log n)$ is established for a single pulsed neuron (that is, for a network consisting of one layer). However, it can also be shown that the VC dimension increases linearly with the number of layers and the number of connections.

Theorem 13 (Maass and Schmitt, 1998; Schmitt, 1999) *For each $L, C \geq 1$, where $C \geq kL$ for some constant k, there exists a multi-layer pulsed neural network with L layers, C connections and binary coding of the inputs that has VC dimension $\Omega(CL)$. This even holds if only the delays are adaptable and all weights and thresholds remain fixed.*

We note that this lower bound for pulsed neural networks in terms of the number of connections and layers equals the best lower bound that is known for sigmoidal networks in terms of the number of weights and layers (Bartlett et al., 1998). Furthermore, the adaptable delay parameters that are used in the construction of (Maass and Schmitt, 1998) for the proof of Theorem 13 can be restricted to a single layer.

Corollary 14 (Maass and Schmitt, 1998; Schmitt, 1999) *It is impossible to give an upper bound for the VC dimension of a multi-layer pulsed neural network with fixed weights solely in terms of the number of connections and layers that have adaptable delays.*

ON THE COMPUTATIONAL COMPLEXITY OF LEARNING WITH ADAPTABLE DELAYS

The issues of learning in pulsed neural networks that we have discussed so far do not suppose the existence of any algorithm: They are related to the informational complexity of learning. We now address the question of computational complexity: How efficient can a learning algorithm be when it uses pulsed neural networks with adaptive delays as hypotheses? And what happens if both delays and weights are adjustable? We employ the theory of NP-completeness to approach this question (Garey and Johnson, 1979).

A basic problem frequently studied in the context of learnability is the so-called *consistency problem*, which for a specified hypothesis class reads as follows: Given a set of labelled examples, does there exist a hypothesis in the class that classifies all examples correctly? At first sight, the consistency problem looks somewhat artificial since it does not take into account the case of errors on the training examples. However, if there is no algorithm that solves the consistency problem and runs in polynomial time for a given hypothesis class then there cannot exist a polynomial-time algorithm for solving the practically more relevant problem of error minimization on the training examples. Therefore, according to the theory of NP-completeness, efficient algorithms that minimize errors for a hypothesis class are unlikely to exist when the consistency problem for this class is NP-complete. The following result shows that this might be the case for a single pulsed neuron.

Theorem 15 *For each $k \geq 2$, the consistency problem for a pulsed neuron with adaptable weights and delays, where the delays take on at most k different values, is NP-complete. This holds also for a pulsed neuron where only the delays are adaptable.*

For simplicity, this result was shown in (Maass and Schmitt, 1997) for a pulsed neuron which may use natural numbers $0, \ldots, k-1$ as delay values. However, using the techniques from this proof the result can easily be extended to any k-element set.

Again, as in the case of the VC dimension, the hardness result shows that learning gets significantly more complex when adaptability of delays comes into play. In contrast, the consistency problem for a McCulloch-Pitts neuron (which for the Boolean domain is equivalent to a pulsed neuron with all delays being equal) can be solved in polynomial time. This discrepancy can be interpreted as saying that adapting weights is easier than adapting delays—at least if single pulsed neurons are considered.

Thus far, there are no results known about the complexity of delay adaptation in networks. However, due to the predominance of hardness results for weight adaptation in neural networks we expect that delay adaptation becomes even harder in networks in some sense yet to be explored.

CONCLUSIONS

We have presented and discussed the latest and most noticeable results on the complexity of computing and learning for pulsed neural networks that use the timing of single spikes to represent information. Concerning the computational power we have seen that a single pulsed neuron can compute functions that when computed by threshold gates require at least a network of linear size. Further, a pulsed neuron is capable of computing functions that cannot be implemented by a higher-order neuron with fixed degree. Concerning the learning capabilities of a pulsed neuron its sample complexity turned out to be significantly larger than for a threshold gate or a sigmoidal gate. For pulsed networks lower bounds for this sample complexity were established that are equal to the known lower bounds for sigmoidal networks. Finally, we concluded that tuning the delays of a pulsed neuron is a task that is believed not to have efficient algorithms with provable performance guarantee.

Although the neuron model we have considered is, compared to biological neurons, still quite simple it suffices to illustrate that the computational and learning capabilities of neurons and networks are considerably enlarged when transmission delays become adjustable. A compelling issue is therefore to study, when further parameters are included in these models in order to get them closer to biology, to what extent their computational and learning abilities are affected by the adaptability of such parameters.

The results presented in this paper show that the use of biological neuron models in machine learning raises issues that did not exist in the conventional networks that have been used so far. Biological neuron models have become attractive for biology and neuroscience since they help to understand natural neural systems. For computer science it is now a challenging task to find algorithms for learning in these models that once will become as popular as those algorithms that have been developed for machine learning in traditional artificial network models.

Acknowledgement. I would like to thank the anonymous referees for helpful comments.

REFERENCES

Agmon-Snir, H. and Segev, I. (1993). Signal delay and input synchronization in passive dendritic structures. *Journal of Neurophysiology*, 70:2066–2085.

Bartlett, P. L., Maiorov, V., and Meir, R. (1998). Almost linear VC dimension bounds for piecewise polynomial networks. *Neural Computation*, 10:2159–2173.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36:929–965.

Bodenhausen, U. and Waibel, A. (1991). The tempo 2 algorithm: Adjusting time-delays by supervised learning. In Lippmann, R. P., Moody, J. E., and Touretzky, D. S., editors, *Advances in Neural Information Processing Systems 3*, pages 155–161, Morgan Kaufmann, San Mateo, Calif.

Eurich, C. W., Cowan, J. D., and Milton, J. G. (1997). Hebbian delay adaptation in a network of integrate-and-fire neurons. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., editors, *Proceedings of the 7th International Conference on Artificial Neural Networks - ICANN '97*, volume 1327 of *Lecture Notes in Computer Science*, pages 157–162, Spinger, Berlin.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, San Francisco.

Gerstner, W. (1999). Spiking neurons. In Maass, W. and Bishop, C. M., editors, *Pulsed Neural Networks*, chapter 1, pages 3–54. MIT Press, Cambridge, Mass.

Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78.

Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1999). Hebbian learning of pulse timing in the barn owl auditory system. In Maass, W. and Bishop, C. M., editors, *Pulsed Neural Networks*, chapter 14, pages 353–377. MIT Press, Cambridge, Mass.

Hopfield, J. J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376:33–36.

Hüning, H., Glünder, H., and Palm, G. (1998). Synaptic delay learning in pulse-coupled neurons. *Neural Computation*, 10:555–565.

Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons.* Oxford University Press, New York.

Koch, C. and Poggio, T. (1992). Multiplying with synapses and neurons. In McKenna, T., Davis, J., and Zornetzer, S., editors, *Single Neuron Computation*, chapter 12, pages 315–345. Academic Press, Boston, Mass.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10:1659–1671.

Maass, W. (1999). Computing with spiking neurons. In Maass, W. and Bishop, C. M., editors, *Pulsed Neural Networks*, chapter 2, pages 55–85. MIT Press, Cambridge, Mass.

Maass, W. and Bishop, C. M., editors (1999). *Pulsed Neural Networks.* MIT Press, Cambridge, Mass.

Maass, W. and Schmitt, M. (1997). On the complexity of learning for a spiking neuron. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 54–61, ACM, New York.

Maass, W. and Schmitt, M. (1998). On the complexity of computing and learning with networks of spiking neurons. In *Electronic Proceedings of the Fifth International Symposium on Artificial Intelligence and Mathematics*, http://rutcor.rutgers.edu/~amai/. Journal version to appear in *Information and Computation.*

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

Murray, A. F. (1999). Pulse-based computation in VLSI neural networks. In Maass, W. and Bishop, C. M., editors, *Pulsed Neural Networks*, chapter 3, pages 87–109. MIT Press, Cambridge, Mass.

Natschläger, T. and Ruf, B. (1998). Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems*, 9:319–332.

Recce, M. (1999). Encoding information in neuronal activity. In Maass, W. and Bishop, C. M., editors, *Pulsed Neural Networks*, chapter 4, pages 111–131. MIT Press, Cambridge, Mass.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W. (1997). *SPIKES: Exploring the Neural Code.* MIT Press, Cambridge, Mass.

Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

Schmitt, M. (1998). On computing Boolean functions by a spiking neuron. *Annals of Mathematics and Artificial Intelligence*, 24:181–191. Extended abstract in *Proceedings of the 8th International Conference on Artificial Neural Networks ICANN 98.*

Schmitt, M. (1999). VC dimension bounds for networks of spiking neurons. In *Proceedings of the 7th European Symposium on Artificial Neural Networks ESANN 99.*

Softky, W. and Koch, C. (1995). Single-cell models. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, pages 879–884. MIT Press, Cambridge, Mass.

Steuber, V. and Willshaw, D. J. (1997). How a single Purkinje cell could learn the adaptive timing of the classically conditioned eye-blink response. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., editors, *Proceedings of the 7th International Conference on Artificial Neural Networks - ICANN '97*, volume 1327 of *Lecture Notes in Computer Science*, pages 115–120, Spinger, Berlin.

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381:520–522.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27:1134–1142.

Vidyasagar, M. (1997). *A Theory of Learning and Generalization.* Communications and Control Engineering Series. Springer, London.

Zador, A. M. and Pearlmutter, B. A. (1996). VC dimension of an integrate-and-fire neuron model. *Neural Computation*, 8:611–624.